Fuzzy Robot Meta-Control

SYSTEM REQUIREMENTS

To run the robotarm demonstration, you need:

- An IBM-PC or compatible.
- A VGA adapter and monitor.
- 512K of memory.
- DOS 3.3 or higher.

INSTALLATION AND EXECUTION

To install the robotarm demonstration onto your hard disk,
do the following:

- Make a directory called robotarm (type "md robotarm").
- Go into that directory (type "cd robotarm").
- Copy the archive into the directory.
- Extract the files from the archive (type "pkunzip robotarm.zip").
- Run the demonstration (type "robotarm").

OVERVIEW

This demonstration addresses the problem of adaptive
trajectory generation for a robot arm. Conventional
trajectory generation involves computing a path in real time
to minimize a performance measure such as expended energy.
This method can be computationally intensive, and it may
yield poor results if the trajectory is weakly constrained.
Typically some implicit constraints are known, but cannot be
encoded analytically.

The alternative approach used in this demo is to formulate
domain-specific knowledge, including implicit and ill-
defined constraints, in terms of fuzzy rules. These rules
utilize linguistic terms to relate input variables to output
variables. Since the fuzzy rulebase is determined off-line,
only high-level, computationally light processing is
required in real time.

Potential applications for adaptive trajectory generation
include missile guidance and various sophisticated robot
control tasks, such as automotive assembly, high speed
electrical parts insertion, stepper alignment, and motion
control for high speed parcel transfer systems.

In this simulation, a robot arm moving along an adaptively
generated trajectory intercepts and grapples a puck. The
user first positions the arm by selecting one of four setup
commands. Then he selects attributes of the puck trajectory
from a menu. When the puck is released, the robot arm moves
to intercept it.

At each sampling interval, a fuzzy expert system evaluates
three classical guidance algorithms, rating their
effectiveness for the current situation. The most effective
algorithm then determines the arm's new heading. The user
sees which algorithm is executing at all times, because a
list of guidance methods is displayed with the current
method highlighted in red.

After the arm intercepts and catches the puck, the

simulation freezes until the user presses a key. Then the
puck disappears, and the arm remains idle until the next
user command. At this point the user can select the Trails
command to see the arm and puck trajectories. All of the
available user commands are described in the section "User
Commands", as well as on a run-time help menu.

USER INTERFACE

The user interface consists of a command buttons window on
the simulation screen and a puck trajectory menu, which
appears when the Pucks command is executed.

SIMULATION SCREEN

The simulation screen contains six windows, as shown in
Figure 3.1.

```
+----------------------------------------------------------------+
| +----------------------------------------+  +--------------+ | |
| |                                        |  |              | | |
| |                                        |  |     Radar    | | |
| |                                        |  |    Window    | | |
| |                                        |  |              | | |
| |                                        |  |              | | |
| |                                        |  |              | | |
| |                                        |  +--------------+ | |
| |                                        |  +--------------+ | |
| |                                        |  |              | | |
| |                 Main                   |  |              | | |
| |                Window                  |  |   Guidance   | | |
| |                                        |  |    Window    | | |
| |                                        |  |              | | |
| |                                        |  |              | | |
| |                                        |  |              | | |
| |                                        |  +--------------+ | |
| |                                        |  +--------------+ | |
| |                                        |  | Puck Status  | | |
| |                                        |  |    Window    | | |
| |                                        |  +--------------+ | |
| |                                        |  +--------------+ | |
| |                                        |  |  Arm Mode    | | |
| |                                        |  |   Window     | | |
| +----------------------------------------+  +--------------+ | |
| +----------------------------------------------------------+ | |
| |                                                          | | |
| |                                                          | | |
| |           Command Buttons Window                         | | |
| |                                                          | | |
| +----------------------------------------------------------+ | |
+----------------------------------------------------------------+
```
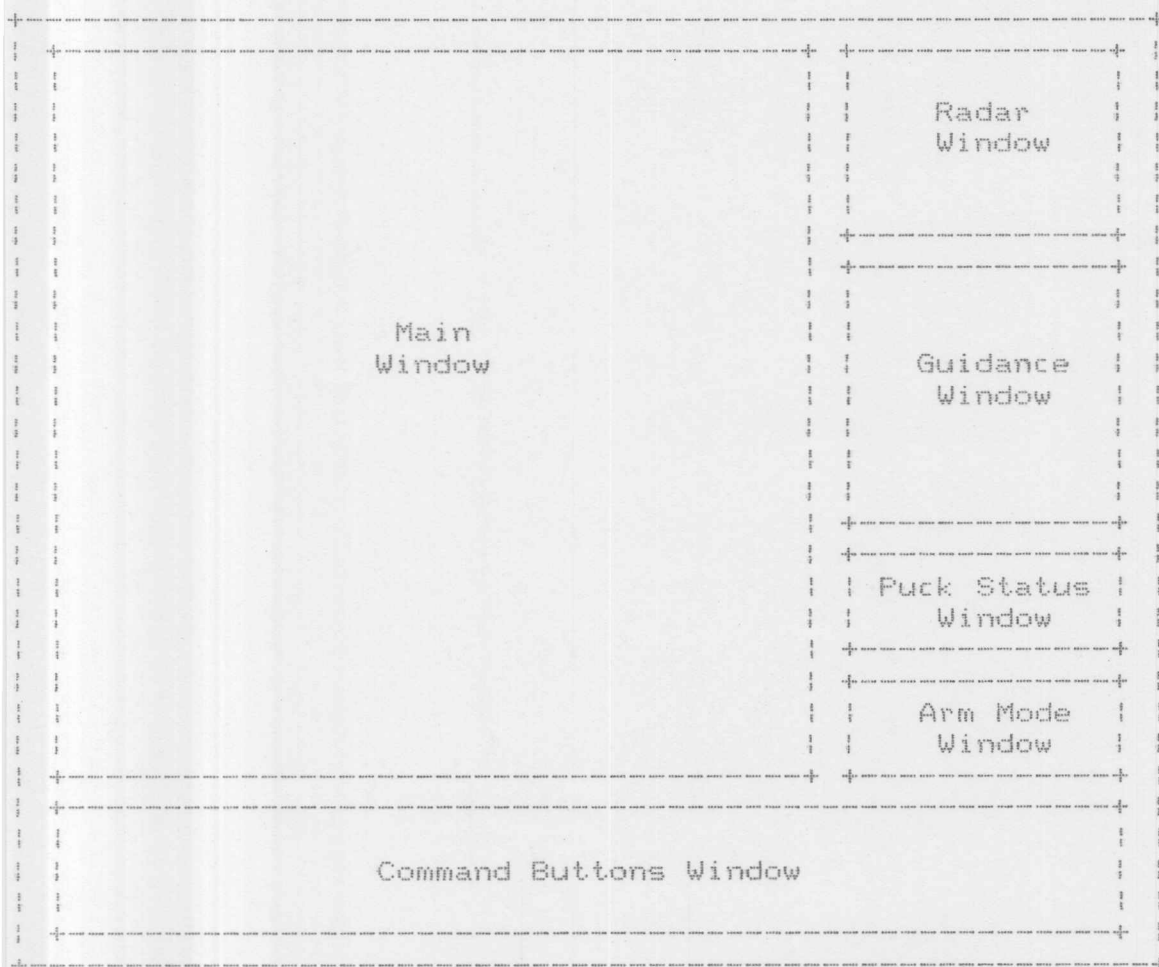
Figure 3.1 - The six windows on the simulation screen.

1) Main Window - The robot arm appears with its base at the
    bottom center of the window. The effector at the end of
    the arm always points in the direction of movement and
    remains fully extended until a puck comes within
    catching range. When the arm heading changes suddenly,
    the effector will turn toward the new heading at its
    maximum turn rate of 5 degrees per second. The current
    arm heading and speed are printed in red just to the
    right of the effector.

    When a puck is launched, it appears as a green ball with
    a red border. The puck's heading and speed are printed
    in cyan immediately to its right. All arm and puck

headings are in the range -180 to 180 degrees, where 0 degrees is horizontal. A set of axes with 0 degrees labeled remains in the lower right corner of the window at all times.

2)   Radar Window - The radar window is a scaled down version of the main window with a few additional features. The maximum range of the robot arm is indicated by a red arc. Pucks are visible as soon as they are launched, even before they appear in the main window. Also, the arm and puck trails are visible whenever the arm is active.

3)   Guidance Window - The guidance window displays the four guidance algorithms. The active algorithm is highlighted in red, while the others are printed in yellow. When the puck is not in use, Arm Test Demo is highlighted.

4)   Puck Status Window - The puck status window shows whether or not a puck has been launched.

5)   Arm Mode Window - The arm mode window displays one of eight possible operational modes. Four are seek modes, including Seek Left Base, Home, Right Base, and User X,Y. The other four are Exercise, Intercept, Catch, and Arm Idle.

6)   Command Buttons Window - The command buttons window displays the ten user commands as rectangular buttons. A red border marks the currently selected button.

   USER COMMANDS

Ten user commands are provided in the command buttons window. They are selected with the arrow keys and activated by pressing ENTER.

1)   Go Left - Moves the arm to its left base position and points the effector at 45 degrees.

2)   Go Home - Extends the arm vertically and points the effector at 90 degrees.

3)   Go Right - Moves the arm to its right base position and points the effector at 135 degrees.

4)   Goto XY - Moves the arm to a user-specified location. Red crosshairs indicate the arm's destination.   The arrow keys position the crosshairs, and the ENTER key starts the arm toward its new destination.

5)   Exercise - Continually moves the arm to random destinations.

6)   Pucks - Displays a puck trajectory menu. The user selects puck speed, path and origin from the available options. ENTER launches the puck. ESCape returns to the simulation screen without releasing a puck.

7)   Trails - Shows the puck trail in yellow and the arm trail in green. The trails appear in both the main window and the radar window.

8)   Pause - Pauses the simulation until another key is pressed.

9) Help - Displays a help menu with a list of user commands
    and a description of the four guidance algorithms.

10) Quit - Ends the simulation.

 PUCK TRAJECTORY MENU

A puck trajectory menu appears when the Pucks command is
executed. The TAB key selects the current field, while the
up and down arrow keys select an option within the current
field.

The first field specifies the puck speed. Slow, medium and
fast speeds are 40, 60 and 90 mm/s, respectively. Variable
speed pucks begin at medium speed, and for every iteration a
random number between -10 and 10 is added to the previous
puck speed. The second field specifies an arced or straight
path. The third field determines the puck's starting
position. Three locations on either side of the screen are
possible.

GUIDANCE ALGORITHMS

While in intercept mode the robot arm is controlled by one
of four guidance algorithms.

1) Constant Bearing - The arm speed and puck velocity
    vector are known. Assuming no change in these
    parameters, the sine law determines the arm heading of a
    straight intercept course.

2) Proportional - The change in arm heading is proportional
    to the change in the line-of-sight angle.

3) Optimal - The change in arm heading is proportional to
    the change in distance times the change in the line-of-
    sight angle.

4) Line of Sight - The arm heading equals the line-of-sight
    angle.

At each sampling interval, a fuzzy expert system determines
an effectiveness rating between -1.0 (not at all effective)
and 1.0 (extremely effective) for each of the first three
algorithms. The most effective algorithm is then used to
update the arm heading. If all three algorithms are rated
below a threshold of -.65, then Line of Sight is executed to
orient the arm.

FUZZY ALGORITHM SELECTION

Three variables are needed to assess each guidance
algorithm's effectiveness. Alpha is the angle formed by the
line of sight and the arm's velocity vector. Beta is the
angle formed by the line of sight and the puck's velocity
vector. Only the magnitude of these angles is considered.
The third input variable is the distance between the
effector and the puck.

 FUZZY SYSTEM OVERVIEW

The fuzzy expert system mentioned in section 4.0 was
developed using Togai InfraLogic's CASE tool, TILShell.   The

Figure 5.1 shows the membership functions (or fuzzy sets)
for Alpha. For each input variable, four membership
functions are defined: Very Small (VS), Small (SML), Medium
(MED), and Big.

```
B   1.00 | \             /\            /\           /-----------------------|
e        |    \       /     \      /     \      /                           |
l        | VS    \  SML      \  MED      \        BIG                       |
i        |    /     \      /     \      /     \                             |
e        | /           \/           \/           \                         |
f   0.00 ------------------------------------------------------//-------
         0         30          60          90         120         180
                            Alpha, degrees
```

             Figure 5.1 - Alpha membership functions.

Consider a particular input value, say Alpha = 40 degrees.
The belief level for each Alpha membership function at the
point Alpha = 40 is the degree to which Alpha is a member of
that fuzzy set. For example, when Alpha = 40 its degree of
membership in all four fuzzy sets can be described as
follows.

                 (Alpha is VS)   to degree 0.00
                 (Alpha is SML) to degree 0.67
                 (Alpha is MED) to degree 0.33
                 (Alpha is BIG) to degree 0:00

Notice in Figure 5.1 that Alpha values greater than 90
degrees are considered Big to degree 1.0. As long as the
user sets up the arm to the correct side before releasing a
puck, Alpha will rarely exceed 90 degrees. Hence, the
membership functions can be compressed toward smaller values
to give finer control.

  OUTPUT MEMBERSHIP FUNCTIONS

All three outputs are described by the same set of
membership functions, as shown in Figure 5.2.

```
B   1.00 |----------------\         /\          /\         /----------------|
e        |                 \      /    \      /    \      /                  |
l        |         NE       \  MLE  \  E     \         VE                    |
i        |                 /    \      /    \      /    \                    |
e        |                /      \/          \/          \                  |
f   0.00 ----------------------------------------------------------------
         -1.0        -0.6        -0.2        0.2         0.6         1.0
                            Effectiveness
```

           Figure 5.2 - Output membership functions.

From left to right, the membership functions are Not
Effective (NE), More or Less Effective (MLE), Effective (E),
and Very Effective (VE).

  FUZZY RULES

Each rule in the fuzzy rulebase is of the form

IF (Alpha is SML) AND (Beta is BIG) AND (Distance is SML)
        THEN Constant Bearing is NE,

Optimal is MLE,
and Proportional is E.

This rule has three outputs, each of which is a fuzzy set.
All three output fuzzy sets are scaled by the belief level
of the premise, also known as the output activation level.
The belief level of each antecedent is determined by the
input membership functions, as described in section 5.3.
Since the antecedents are combined with AND, the output
activation level is simply the minimum of the three
antecedent belief levels.

For each of the three output variables -- Constant Bearing,
Optimal, and Proportional -- the scaled output fuzzy sets of
all the rules are summed to get a combined output fuzzy set.
The centroid of this combined output fuzzy set is the
effectiveness rating for that particular algorithm.

CATCH MODE

As the arm and puck approach an interception point, the arm
enters Catch mode. During the next six time intervals, the
effector closes, and the arm slows down from its maximum
speed of 60 mm/s to approximately zero at the point of
contact. During this slow down phase, the fuzzy expert
system still determines the most effective guidance
algorithm, although the arm speed is controlled by a
different equation.

After the slow down phase, the arm and puck both move along
the puck's path, slowing down over four time intervals.
When they have stopped, the simulation freezes until any key
is pressed. This apparent catching motion was included
strictly for aesthetic appeal. The fuzzy expert system does
not operate after contact is made with the puck.

CONCLUSIONS

The robot arm trajectory simulation clearly demonstrates
that domain-specific knowledge can be encoded off-line as a
set of linguistic fuzzy rules. Using only these rules and
membership functions to define the linguistic terms, the
fuzzy expert system can quickly decide which guidance
algorithm is most effective under the current circumstances.

Togai InfraLogic currently markets a chip called the FC110
designed for efficient processing of fuzzy rules. For the
fuzzy system used in this simulation -- 3 inputs, 3 outputs,
and 33 rules -- the FC110 operating at 20 MHz can return
crisp effectiveness outputs in 315 ms. Since the FC110 is a
general purpose microprocessor, a Single Board Fuzzy
Computer (SBFC) utilizing this chip could control a robot
arm in real time with no need for additional computer power.

FURTHER INFORMATION

Please direct any comments, suggestions, or questions to

    Togai InfraLogic, Inc.
    5 Vanderbilt
    Irvine, CA 92718-2011
    Phone: (714) 975-8522
    F A X: (714) 975-8524